Vektskål

Du har en vekt som har to skåler som er koblet sammen med en vektarm. Dersom innholdet i de to skålene veier like mye er vekten i balanse. Dersom innholdet i den ene skålen veier mer enn i den andre så vil vekten helle mot den tyngste siden.

For å veie ut en viss vekt så har du en rekke vektlodd. Dersom du f.eks. skal måle opp 40 gram med mel kan du plassere lodd med samlet vekt på 40 gram i den ene skålen og deretter helle mel i den andre skålen til de er i balanse. Noen ganger skjer det at du ikke har de rette vektene for å måle opp en mengde bare ved å plassere de på den ene skålen. Da er en løsning å plassere noen av vektene sammen med varen som skal måles.

Du ønksker å vite hvordan du skal kunne veie opp en viss mengde.

Input

Første linje inneholder to heltall: X (antall gram du skal måle opp) og N (antall lodd du har tilgjengelig).

Neste linje inneholder N forskjellige heltall L_i (for $i \in \{1...n\}$) skilt med mellomrom. Disse er vektene av loddene du har, gitt i gram.

Loddene kommer alltid i stigende rekkefølge etter vekt. Det letteste loddet vil alltid veie 1 gram, og alle de andre loddene vil veie maksimalt 3 ganger så mye som det forrige.

Output

Output skal beskrive hvordan loddene skal plasseres.

Første linje skal være et heltall U, antall lodd i venstre vektskål. Andre linje skal inneholde U heltall, vektene av de loddene som er i venstre vektskål.

Tredje linje skal være et heltall V, antall lodd i høyre vektskål. Fjerde linje skal inneholde V heltall, vektene av de loddene som er i høyre vektskål.

Summen av vekten av loddene i venstre vektskål skal være nøyaktig X mer enn summen av vekten av loddene i høyre vektskål. Det vil alltid være minst en løsning. Dersom det er flere løsninger, oppgi hvilken som helst av de. Merk at hvert lodd kan brukes maksimalt en gang.

Begrensninger

$1 \leq X \leq 1\,000\,000\,000\,000\,000\,000$

$1 \leq N \leq 60$

$1 \leq L_i \leq 1\,000\,000\,000\,000\,000\,000$ for alle i

I testsett verdt 50 poeng vil X og alle $L_i \leq 10\,000$

Time limit: 2 s.

Merk at tallene kan være større enn de får plass til i en 32 bits heltall. Bruk derfor 'long long' i C++ eller 'long' i Java.

Eksempler

Input	Output

`16 4`
`1 3 9 20`
	`1`
	`20`
	`2`
	`1 3`

Input	Output	Kommentarer

`121 7`
`1 2 5 10 25 50 100`
	`2`
	`100 25 1`
	`1`
	`5`
		Her finnes det flere mulige lønsinger

Teleportør

Mange dataspill utspiller seg i store, åpne verdener hvor man kan reise fritt mellom mange store byer. Dette er spennende i ca. 5 minutter før man blir lei av å vandre rundt og heller skulle ønske seg at det var en raskere måte å komme seg fra by til by. Derfor har slike spill ofte en mulighet for å kunne hoppe mellom en del steder, så fort du har oppdaget de.

Verdenen som spillet vi skal se på nå inneholder N byer, nummerert fra 0 til $N-1$, som er forbundet med veier. I M av byene så finnes det en teleportør. Når du kommer til en by med en teleportør så blir den åpnet opp for deg. Befinner du deg i en by med en teleportør kan du reise til en hvilken som helst annen by hvor det er en teleportør som du har åpnet opp uten at det tar noe tid.

Når spillet begynner befinner du deg i byen med nummer 0. Du ønsker å åpne opp alle teleportørene på kortest mulig tid. Hvor lang tid vil dette ta deg?

Input

Første linje inneholder tre heltall N (antall byer), M (antall byer med teleportører) og K (antall veier). Neste linje inneholder M forskjellige heltall, som er nummerene på de byene som inneholder teleportører. Deretter følger K linjer på formen A_i B_i T_i, som hver linje beskriver en av veiene. En slik linje betyr at det er en vei med endepunkter i byene med nummer A_i og B_i, og som tar T_i minutter å reise.

Output

Et heltall - det minste antall minutter det tar før du kan ha åpnet opp alle teleporørene.

Begrensninger

$2 \leq N \leq 5\,000$

$2 \leq M \leq 50$

$1 \leq K \leq 20\,000$

$1 \leq T_i \leq 1\,000$

Time limit: 2 s.

Testsettgruppe	Poeng	Ytlingere begrensinger
Gruppe 1	30	$N = M$. (Alle byene har en teleportør)
Gruppe 2	20	By nummer 0 inneholder en teleportør
Gruppe 3	50	Ingen andre begrensninger

Alle testsettene i en gruppe må løses riktig for at gruppen skal gi poeng.

Eksempler

Input	Output

`4 4 5`
`0 1 2 3`
`0 2 55`
`0 1 50`
`1 2 40`
`2 3 60`
`3 0 20`
	`110`

Input	Output

`3 2 2`
`0 1`
`0 1 70`
`0 2 15`
	`70`

Input	Output

`3 2 2`
`1 2`
`0 1 70`
`0 2 15`
	`100`

Klassebilde

N elever har gått i klasse sammen i k år. Hvert år har det blitt tatt et klassebilde, med forskjellig oppstilling hvert år. Du mistenker at noen av elevene ikke liker hverandre noe serlig. Du antar at elever som ikke liker hverandre stort sett prøver å stille seg så lengst mulig fra hverandre på klassebildene. For å finne ut hvilket par av elever som liker hverandre minst så vil du finne ut hvilke to elever som i gjennomsnitt står lengst unna fra hverandre på klassebildene.

Input

Først en linje med to heltall $2 \leq N \leq 50\,000$ og $1 \leq k \leq 6$ adskilt med mellomrom. Deretter kommer det k linjer, der den i'ende linjen av disse beskriver oppstillingen på det i'ete bildet. Hver elev har et unikt ID-nummer fra 1 til og med N. Hver linje som beskriver et klassebilde inneholder alle tallene fra 1 til N, adskilt med mellomrom, i samme rekkefølge som elevene står oppstilt på klassebildet.

Output

To tall a og b, med $a < b$, på en linje, adskilt med mellomrom. Dette skal være ID-nummerene til de to elevene som liker hverandre minst. Dersom det er flere mulige svar, skriv ut det paret med minst a. Dersom det fremdeles er flere mulige svar, skriv ut det av disse med minst b.

Begrensninger

$2 \leq N \leq 50\,000$

$1 \leq K \leq 6$

Time limit: 4 s.

Testsettgruppe	Poeng	Ytlingere begrensinger
Gruppe 1	22	$n \leq 50$
Gruppe 1	23	$n \leq 800$
Gruppe 2	9	$k = 1$
Gruppe 3	46	Ingen andre begrensninger

Alle testsettene i en gruppe må løses riktig for at gruppen skal gi poeng.

Eksempler

Input	Output	Kommentarer

`10 2`
`1 2 3 4 5 6 7 8 9 10`
`5 3 4 2 1 10 9 7 8 6`
	`3 8`	Barn 3 og 8 har en avstand på 5 i det første klassebildet og en avstand på 7 i det andre klassebildet for en gjenomsnitling avstand på 6.

Input	Output

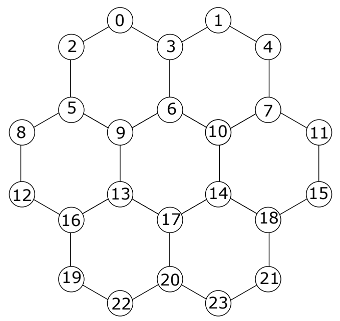
`12 3`
`8 5 9 1 10 3 2 12 11 7 4 6`
`7 11 4 1 10 8 5 6 9 12 2 3`
`7 10 5 1 6 4 12 8 11 3 9 2`
	`7 9`

Bosetterene i Bergen

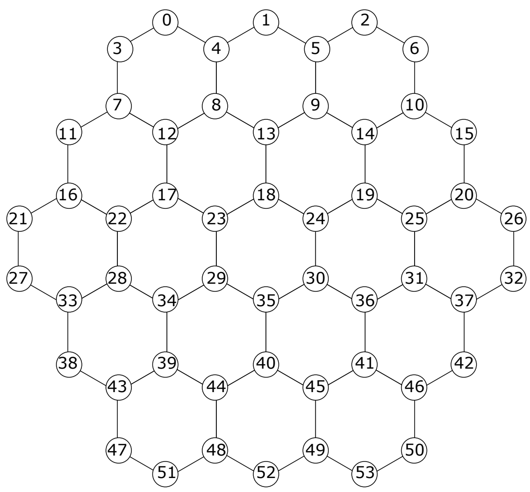
Bosetterene i Bergen er et brettspill som går ut på å bygge flest mulig hus på et brett. Brettet er satt sammen av regulære sekskanter på følgende vis:

- På øverste rad legger man n sekskanter inntill hverandre, slik at alle har en spiss pekende opp.
- Inntill disse, på raden under, legger man $n + 1$ sekskanter, fortsatt med en spiss pekende opp.
- Gjenta å legge ut flere rader, hver med 1 sekskant mer enn den forrige, helt til du har lagt ut totalt n rader.
- Legg deretter ut $n - 1$ rader til, men nå hvor hver rad har 1 sekskant mindre enn den forrige. Siste raden har dermed n sekskanter.

Hjørnene på brettet er alle punkter som er et hjørne i minst en av sekskantene. Disse er numerert ovenifra og nedover, og innenfor en rad fra venstre til høyre, fra 0 og oppover. Skissene under viser hvordan brettet ser ut for $n = 2$ og $n = 3$, og hvordan hjørnene er numerert.



n=2



n=3

Det er kun i disse nummererte hjørnene som det er mulig å bygge hus og det kan kun være ett hus per hjørne. I tillegg er det ikke lov å bygge et hus på et hjørne som er nabo til et annet hjørne hvor det befinner seg et hus. F.eks. hvis det er et hus på hjørne 9 i brettet hvor n = 2, så kan man ikke bygge hus på noen av hjørnene 5, 6 eller 13 (men huset på hjørne 9 forhidrer deg ikke i å bygge på f.eks. hjørne 10 eller 14).

Du har allerede plassert ut en del hus på brettet. Nå lurer du på hvordan mange flere hus det er mulig å bygge med reglene gitt over.

I denne oppgaven skiller vi mellom *korrekte* og *optimale* løsninger. En løsning er *korrekt* dersom den beskriver en gyldig plassering av hus. En løsning er *optimal* hvis den er korrekt og det ikke finnes noen korrekt løsning som inneholder flere hus.

Input

Første linje inneholder et heltall T - testsettgruppen for dette testsettet. Se beskrivelse under for hvordan løsnigner blir vurdert i forskjellige testsettgrupper. Andre linje inneholder et heltall n (størrelsen på brettet slik definert over) og k (antall hus allerede bygd). Deretter følger k linjer, hver med forskjellige tall p_i, posisjonen på et av husene som allerede har blitt bygd.

Output

Første linje skal inneholde et tall V, antall hus i løsningen din. De neste V linjene skal hver inneholde et heltall Q_i, posisjonen på hus nummer i som du vil bygge.

Løsningen vil alltid være unik.

Begrensninger

Time limit: 2 s.

Testsettgruppe	**Poeng**	**Begrensinger**	**Krav til løsning**
Gruppe 1	15	$n = 2$	Løsningen må være optimal.
Gruppe 2	12	$2 \leq n \leq 14$	Løsningen trenger kun å være korekkt og inneholde minst 1 nytt hus. Her vil alltid være mulig å plassere ut minst 1 nytt hus.
Gruppe 3	33	$2 \leq n \leq 8$	Løsningen må være optimal.
Gruppe 4	40 Mulighet for delvis poengsum	$2 \leq n \leq 14$	Løsningen trenger kun å være korrekt, men poengsummen gitt vil variere ut i fra hvor mange hus løsningene din inkluderer. Dersom de optimale løsningene for testsettene i denne gruppen inneholder totalt O hus, og løsningene dine inneholder totalt V hus, så er poengsummen du får for denne gruppen gitt ved $40 \times 0.995^{(O-V)}$.

Alle testsettene i en gruppe må løses riktig for at gruppen skal gi poeng.

Eksempler

Input	Output	Kommentarer
1 2 3 0 9 14	7 1 7 8 15 16 21 22	Det er flere optimale løsninger her.

Input	Output
1 2 5 5 3 7 18 20	3 12 13 19